A FAIR ALGORITHM IN A MARRIAGE MARKET

Aditya V Kuvalekar *†

Abstract

I propose a new algorithm to obtain a stable matching in a classical two-sided marriage market. The algorithm treats men and women equally and often produces a matching that is not one of the extremes. The algorithm has a novel feature that both sides of the market make proposals in every round. The algorithm does not distinguish (ex-ante) between men and women at any stage. The ex-post distinction arises primarily because in every round potential cycles are formed that are broken arbitrarily. Moreover, the algorithm can be computed in polynomial time and hence, from a practical standpoint, can be used in markets where fairness considerations are important.

JEL Classification: C72, C78, D41

1 Introduction

I study the classical two-sided marriage market due to Gale and Shapley ([4]). In this market there are two sides - Men and Women and they have strict preferences over the other side. A matching is an assignment wherein each man is matched to a woman or himself (indicating that he is unmatched) and vice-versa. Moreover, no two men are assigned a same woman. A desirable property for two-sided marriage markets is *stability.* Stability requires that no subset of men and women would want to deviate and form their own assignment. The question about the existence of a stable matching in marriage market was settled positively in a seminal paper by D.Gale and L.Shapley in 1962 ([4]) by proposing a *Deferred Acceptance Algorithm* (DAA henceforth). The algorithm established that in a two-sided marriage market with strict preferences, there always exists at least one stable matching. Moreover, the authors also prove that the matching obtained through DAA is the best matching for the proposing side amongst the set of stable matchings. Following this work, Knuth (1976)([7]) proved that when all agents have strict preferences the common preferences of two sides of the market are opposed to each other

*I am indebted to Jay Sethuraman for countless discussions and guidance in the development of this algorithm. I sincerely thank Vikram Manjunath for his detailed comments and feedback on the paper. I also wish to thank Christine Cheng, Kevin Dick, Elliot Lipnowski, Kaustubh Nimkar, Efe Ok, Debraj Ray, Parag Pathak, Antonio Romero-Medina, Ennio Stacchetti, Rakesh Vohra and Leeat Yariv for their useful comments.

†New York University. Email: avk255@nyu.edu

on the set of stable matchings. That is, for any two stable matchings, μ_1 and μ_2, all the men **weakly** prefer μ_1 to μ_2 *if and only if* all the women weakly prefer μ_2 to μ_1. Morever, it was also proved that DAA outcome is the best amongst all the stable matchings for the proposing side. As a natural corollary, we obtain that any DAA outcome is not only the best for the proposing side but it is also the worst for the side receiving proposals.

This raises a questions as to can we devise a procedure that does not distinguish between the two sides explicitly and gives a stable match. With this focus, I propose a new algorithm wherein both sides of the market propose in each round to a set of agents. The algorithm always yields a stable matching that is often different from either extremes. Moreover, the algorithm terminates in polynomial time. Since the algorithm does not distinguish between men and women, the natural way to view this proposed procedure is from the perspective of *Ex-Ante Fairness*.

The primary motivation of this algorithm related to fairness in two-sided marriage market. To quote Donald Knuth [7],

> The different algorithms considered until now favour the men, and if we interchange the roles of men and women they would become favorable to the women. Such injustice is too shocking for the present day. Can we therefore find a solution that treats both sexes fairly?
>
> We can enumerate all the stable solutions and choose the most satisfying matching according to certain criteria. This might take a long time if the number of stable matchings is large. We do not know in general if there exists a large or small number of solutions. If therefore seems preferable to use another method.

My algorithm can be described as follows - Every round begins with an existing matching and a set of agents for each man and a woman that indicates the agents each man or woman are willing to accept in that round. In every round all the agents (Men and Women) propose to this set. If an agent is matched, then (s)he proposes to his(her) current match and all the agents better than it. Then, for every agent we determine all those agents who are proposing to him(her). Thus, for every agent, we can determine a set of *mutually proposing agents.* Based on this, we construct a graph with nodes as agents and each agent points to his (her) topmost mutually proposing agent. This may be empty, in which case the outdegree of such agents is 0. Each agent will have at most one outgoing edge from it. Moreover, if an agent has an incoming edge then it must have an outgoing edge as well. The graph constructed will have cycles that are broken arbitrarily (in favour of men or women) to form a temporary matching. The unmatched agents continue to propose to a larger set of agents in the following round by enlarging their set. When this set cannot increase further, an agent is made inactive. Eventully, either all the agents become inactive or a matching is obtained that is stable for the active agents. Then, inactive agents are made active serially. Very often, the algorithm terminates to give a stable matching without any agent ever being inactive. However, the operation of making agents inactive is required to ensure that the algorithm does not cycle if some agents reach the end of their lists.

The reason why this algorithm terminates in polynomial time is because at no stage does the algorithm try to solve for a complex stability problem that increases the computation complexity.

At the same time, the reason why we still obtain stability is because the way the algorithm proceeds, any blocking pairs are eventually discovered to yield a stable matching.

Since at every stage the cycles are broken with a toss of coin, no side of the market enjoys an ex-ante advantage over the other. Given the nature of the procedure wherein agents keep proposing to larger sets, the algorithm often produces a matching that is not one of the two extremes. Moreover, given that the algorithm is computationally efficient, this procedure can be viewed as a strong practical alternative to DAA in markets where fairness considerations seem to be of importance.

2 Existing Literature

It is well-known that the set of stable matching has a specific lattice structure (Roth and Sotomayor, 1990 [12]). and the two DAA pick the two extreme points of this lattice. Masarani and Gokturk (1990)[10] showed impossibilities to obtain a fair stable matching wherein their notion of *fairness* was Rawlsian. While negative, this result demanded that the rule be deterministic. To circumvent this problem one way to recover fairness is to use probabilistic (stable) matching mechanisms that are ex ante fair and/or 'procedurally fair', as in Klaus and Klijn (2006)[5], and Ma (1996)[8].

The other approach towards fairness is based on the outcomes. For example, Romero-Medina (2005)[11] define an *Equitable Set* to capture fairness based on the Rawlsian criterion. In contrast, I follow a procedural approach towards fairness in proposing a new algorithm with both sides proposing in each round. Moreover, as seen in example 5.2, there are instances where the Rawlsian notion picks, as an *Equitable Set*, only one of the extreme matchings while my procedure picks both the extreme.

Ma(1996)[8] proposed a procedure wherein we start with a *Random Priority* over agents. Following the priority, we start with an empty match and add agents one by one to the matching satisfying all the blocking pairs *within the match*. That is, blocking pairs outside the agents not added so far are ignored. While this procedure does not distinguish between sexes it can result in situations where it can select only the extremes but not the middle outcomes. In contrast, the algorithm I propose has a narrower support and in many examples (such as those in Subsection 4.1) it will pick a strict subset of stable matchings that do not involve any of the extreme outcomes due to Gale and Shapley. A starker example highlighting the difference is given in Example 5.3.

Using another approach, Teo and Sethuraman (1998)[15] and Sethuraman et al.(2004)[14] established the existence of a deterministic 'compromising mechanisms' for marriage and college admissions models respectively. Specifically, they showed that if all agents order their (possibly non-distinct) matches at the, say, k stable matchings from best to worst, then the map that assigns to each agent of one side of the market its l-th best match and to each agent of the other side its $(k - l + 1)^{st}$ best match constitutes a stable matching. Teo and Sethuraman (1998)[15] and Sethuraman et al. (2004)[14] used linear programming tools to prove that these '(generalized) median stable matchings' are indeed well-defined and stable. The main issue with the *median stable matching* is that if computing the *median stable matching* can be done efficiently then computing the number of stable matchings can also be done efficiently and the latter is known to be a computationally hard problem. [2]

Cheng et al.(2014) [3] proposed a concept of *Center Stable Matchings* that are stable matchings whose maximum distance to any stable matching is as small as possible. These matchings are close to median stable matchings and can also be computed efficiently. However, primarily *median stable matching* or *center stable matching* are fair due to the properties of the stable matchings themselves. My approach to fairness through the proposed algorithm is the procedural fairness of the mechanism itself. In that sense, my algorithm is more similar to the mechanism due to Ma [8].

For the problem of school choice, when schools carry out their matching independetly, Manjunath and Turhan (2014)[9] propose a new mechanism to avoid wastage of seats wherein, for the wasted seats, they propose a *Matching and Rematching* mechanism that yields a solution different than the two extremes. The outcome of their procedure varies depending on the number of iterations and also is not gender neutral.

The paper is organized as follows. In section 3, we discuss the preliminaries of the model and the setup. This is a standard setup of the classical two-sided marriage market and can be skipped by those familiar with the literature. Section 4 describes the algorithm in detail. Section 5 has a number of examples illustrating the execution of the algorithm along with an example of an instance where the algorithm cycles. Section 6 is devoted to the proofs of the fact that the algorithm always yields a stable matching. Section 8 describes the fairness properties of the algorithm. And the last section (9) is the conclusion.

3 Preliminaries

There are two finite disjoint sets of agents, call them *Men* and *Women*.

$M = \{m_1, m_2, \ldots, m_l\}$ is a set of men.

$W = \{w_1, w_2, \ldots, w_k\}$ is a set of women.

We denote a generic agent by i. If we need to distinguish between a man and a woman for illustration, we will do so by labeling them as m_j and w_l respectively.

Each agent has a linear ordering over the agents on the other side of the market. Moreover, in that ordering one element is the agent himself indicating that the agents is also potentially willing to remain single over being matched to some agents from the other side. In what follows though, we will study a model wherein the agent will always prefer to be matched over remaining single. I do not believe that any results are altered in any way by relaxing this assumption. However, that needs to be proved in general.

Man m's preferences can be represented as $\succeq_m$. For a generic agent, the ordering will be denoted by $\succ_i$.

For example, $\succ_i = (1, 3, 4, 7, 5, 8, 2, 6)$ means $1 \succ_i 3 \succ_i 4 \succ_i 7 \ldots \succ_i 2 \succ_i 6$.

In general, to avoid distinguishing between men and women in terms of notation, we will denote by P_i the preferences of agent i over the other agents.

$$P := (P_i)_{i \in M \cup W}$$

Let P_M and P_W denote the preferences of Men and Women respectively.

P_i^j denotes the j^{th} ranked element according to the preferences of i.

Let $\mathcal{P}$ denote the class of all preference profiles.

Lastly, let A_i denote the set of acceptable agents for each i. That is, i would prefer being matched to any agent in i over being single, and would prefer being single than to match with any agent not in A_i.

A marriage market is a triple (M, W, P). A matching is a one-one map $\mu : M \cup W \to M \cup W$ such that $\mu(m) \in W \cup \{m\}\ \forall\ m$ and $\mu(w) \in W \cup \{w\}\ \forall\ w$. Moreover, for the matching to be meaningful we also require that $\mu(m) = w$ iff $\mu(w) = m$. In general, $\mu(i) = j \Leftrightarrow \mu(j) = i$.

Let $\mathcal{M}$ be the set of all feasible matchings.

A desirable property for a matching is that it be stable.

Definition 3.1. *A matching μ is stable if $\nexists\ (i, j)$ such that $j \succ_i \mu(i)$ and $i \succ_j \mu(j)$. Moreover, such pair(s) (i, j) are called as blocking pair(s) in a matching μ.*

Through their algorithm Gale-Shapley (1962) established that in every marriage market there always exists a stable matching. Let $S(P)$ denote a set of stable matchings for the market (M, W, P).

Define an ordering $\succ_M$ over $S(P)$ as follows -

$$\mu \succ_M \hat{\mu}(\neq \mu) \Leftrightarrow\ \mu(m) \succeq_m \hat{\mu}_m,\ \forall m \in M.$$

Using this ordering, we can define $\lambda := \mu \vee_M \mu'$ as a function on N that assigns to each man his most preferred match from μ and μ'. Formally,

$$\lambda(m) = \begin{cases} \mu(m) & \text{if } \mu(m) \succ_m \mu'(m) \\ \mu'(m) & \text{otherwise} \end{cases} \tag{3.1}$$

$$\lambda(w) = \begin{cases} \mu(w) & \text{if } \mu'(w) \succ_m \mu(w) \\ \mu'(w) & \text{otherwise} \end{cases} \tag{3.2}$$

$\mu \wedge_M \mu'$ are also defined analogously by assigning each man his worst preferred mate amongst μ and μ' and converse for the women.

The following theorem due to Conway established that the set of stable matchings is a lattice. In fact, it is a distributive lattice.

Theorem 3.1. *If $\mu, \mu' \in S(P)$ then, $\mu \vee_M \mu', \mu \wedge_M \mu' \in S(P)$.*

Since the set of stable matchings is finite (obviously) there exists a $\succ_M$ **maximum** element. Simiarly, if we define $\succ_W$ analogously, then there exists a $\succ_W$ **maximum** element according to the order of women. Gale and Shapley [4] proved that the version of their algorithm in which men propose yields a maximum element according $\succ_M$ in $S(P)$. Similarly, a woman proposing version of the algorithm yields a matching that is the most preferred by women.

4 Algorithm

4.1 Examples

A simple example of such a market wherein one could expect to obtain an outcome different than the Gale-Shapley outcome should the two sides be allowed to make proposals is the following.

m_1	m_2	m_3
w_1	w_2	w_3
w_2	w_3	w_1
w_3	w_1	w_2

Here, $1,2,3$ can be thought of as Men and $4,5,6$ can be thought of as Women. The each column gives the preferences of a corresponding agent. That is, $4 >_1 5 >_1 6$, $5 >_2 6 >_2 4$ and so on.

Women's preferences are as below.

w_1	w_2	w_3
m_2	m_3	m_1
m_3	m_1	m_2
m_1	m_2	m_3

Observe that the outcome of the Men Proposing DAA is $\{(m_1,w_1),(m_2,w_2),(m_3,w_3)\}$. Simlarly, the women proposing Gale-Shapley yields, $\{(m_1,w_3),(m_2,w_2),(m_3.w_1)\}$. However, imagine a procedure wherein both sides make proposals each round to a set of agents starting from their top agents. In each round they expand their sets of acceptable agents according to their preferences. And we match agents only when they both list each other as mutually acceptable.

Then, in the first round, m_1 would propose to w_1, m_2 to w_2, m_3 to w_3, w_1 to m_2, w_2 to m_3 and w_3 to m_1. None of the agents find each other mutually acceptable. But in the second round, m_1 proposes to w_1 & w_2. m_2 proposes to w_2 & w_3 and so on. In this round, m_1 and w_2 find each other mutually acceptable. So do m_2 and w_3, and m_3 and w_1. So, we match $(m_1,w_2),(m_2,w_3),(m_3,w_1)$.

4.2 Example 2

To further expand on the idea of both sides making proposals, consider the following preferences of men.

m_1	m_2	m_3	m_4
w_1	w_2	w_3	w_4
w_2	w_3	w_4	w_1
w_3	w_4	w_1	w_2
w_4	w_1	w_2	w_3

Women's preferences are -

w_1	w_2	w_3	w_4
m_2	m_3	m_4	m_1
m_3	m_4	m_1	m_2
m_4	m_1	m_2	m_3
m_1	m_2	m_3	m_4

The Gale-Shapley outcomes are -

$$\{(m_1, w_1), (m_2, w_2), (m_3, w_3), (m_4, w_4)\} \text{ and } \{(m_1, w_4), (m_2, w_1), (m_3, w_2), (m_4, w_3)\}.$$

We will look at this example again more formally but as of now let us look at it informally. Imagine a procedure wherein the agents from both sides of the market make proposals to a set of agents every round. If there is no agent whom they are proposing to and is also proposing them then they expand their *proposing to* set of agents in the following round. They start by proposing to the top agent according to their preferences and expand the set of agents they propose to according to their preferences.

In the first round, all the agents propose to their top choice and no pair finds each other mutually acceptable as if an agent i proposes to j in the first round then j is the top agent according to i and i is the worst agent according to j. In the following round, for example, m_1 proposes to w_1 and w_2. On the other hand, w_1 proposes to m_2 and m_3, while w_2 proposes to m_3 and m_4. m_2 proposes to w_2 and w_3. w_2 proposes to m_3 and m_4, while w_3 proposes to m_4 and m_1. It can be seen that there are no two agents who propose to each other in this round too.

However, in round 3, consider agent m_1. He proposes to w_1,w_2 and w_3. Observe that w_2 proposes to m_3, m_4 and m_1, while w_3 proposes to m_4, m_1 and m_2. That is, there are two agents who m_1 is interested in being matched to and they too are interested in being matched with m_1. Those two agents are w_2 and w_3. m_1 would prefer being matched to w_2 over w_3. But observe that w_2 too has two mutually acceptable agents and she would prefer being matched to m_4 rather than m_1. Continuing this way, if we look at the top agent for each person from the list of their mutually acceptable agents in this round, we obtain a cycle as below.

$$m_1 \to w_2 \to m_4 \to w_1 \to m_3 \to w_4 \to m_2 \to w_3 \to 1$$

Now, if we break this cycle either in favour of men or women we obtain the following two matches -

$$\{(m_1, w_2), (m_2, w_3), (m_3, w_4), (m_4, w_1)\} \text{ or } \{(m_1, w_3), (m_2, w_4), (m_3, w_1), (m_4, w_2)\}$$

The entire algorithm described below makes this simple idea from the above two examples formal and solves a complicated cycling problem that can often arise in markets with arbitrary preferences. But at the core, the motivation and the essence is these simple examples.

4.3 Formal Algorithm

The algorithm, in each round, needs two things. The existing matching μ and an index for every agent that is denoted by β. The index is the number of agents that a person proposes to in his list in a particular round *if (s)he is unmatched.* If the agent is matched, (s)he proposes to only those agents that are better than the his(her) current match. That is, in the algorithm, each agent is proposing to some agent(s) whether matched or unmatched.

Recall that A_i is the set of all acceptable agents for i. β_i denotes the number of agents an agent i looks at in any given round. The algorithm increments β_i for every unmatched agent in every round until it becomes $|A_i| + 1$. After that, the agent becomes inactive and his β isn't increased further. In every round, there will be a set of active agents and a set of inactive agents. Once an agent is inactive, (s)he does not become active until either all the agents are inactive or the set of active agents are matched and the matching does not change between consecutive rounds.

Definition 4.1. *An agents i is active in any round of the algorithm if his (her) $\beta_i \leq |A_i|$.*

Define,

$$\mathcal{A} := \{i \in \mathbb{N}^{|M \cup W|} : \beta_i \leq |A_i| + 1\}$$

That is, $\mathcal{A}$ is the set of feasible β_is. Let $\Gamma := \{i \in M \cup W : \beta_i \leq |A_i|\}$.
Γ is the set of active agents at any given time.

ALGORITHM 1:

---

1. Set $\mu_0 = \emptyset$ and $\beta_i = 1 \forall i \in \{M \cup W\}$.
2. DO
$$(\mu^{k+1}, \beta^{k+1}) = \mathbb{H}(\mu^k, \beta^k)$$
UNTIL $(\mu^{k+1}, \beta^{k+1}) = (\mu^k, \beta^k)$.
3. Let the terminal β^k be called β and the terminal μ^k be called μ. If, $\beta_i \leq |A_i|\ \forall i$ then END. μ is a stable matching. If not, proceed as below.
4. $U = \{j : \beta_j = |A_j| + 1\}$. These are a set of unmatched agents that are excluded. Choose an arbitrary permutation π of agents in U. Input for the following operation is (μ, β), the final matching and β obtained in STEP 2.
5. For $j = 1$ TO $|U|$,
   (a) Set $\beta_{\pi(j)} = |A_{\pi(j)}|$. That is, make the agent $\pi(j)$ active.
   (b) Set $(\mu^1, \beta^1) = (\mu, \beta)$.
   (c) DO
   $$(\mu^{k+1}, \beta^{k+1}) = \hat{\mathbb{H}}(\mu^k, \beta^k)$$
   UNTIL $(\mu^{k+1}, \beta^{k+1}) = (\mu^k, \beta^k)$.

(d) Set $(\mu, \beta) = (\mu^k, \beta^k)$. Increase j and go to step (a) until $j = |U|$.

6. μ obtained from STEP 5 is a stable matching.

Therefore, in order to prove that the algorithm gives a stable match on all profiles we first need to define what maps $\mathbb{H}$ and $\hat{\mathbb{H}}$ are. Thereafter, we will prove that the algorithm outputs a stable match.

4.4 Evaluation of $\mathbb{H}$

Input: μ, β.
Output: Updated $\hat{\mu}, \hat{\beta}$.

1. Set $\hat{\mu}_i = 0\ \forall i \in M \cup W$. Set, $N = \{j : \beta_j \leq |A_j|\}$.

   Set $A = N$ (A is the set of active agents that remains the same in the execution of $\mathbb{H}$.)

2. Define, $\forall i \in N$,

$$S_i := \begin{cases} \{j : j \succ_i \mu(i)\} & \text{If } \mu(i) > 0 \\ \text{Top } \beta_i \text{ elements in } i\text{'s preference list} & \text{If } \beta_i \leq |A_i| \text{ and } \mu(i) = 0 \\ \emptyset & \text{Otherwise} \end{cases} \tag{4.1}$$

3. Define, $\forall i \in N$,

$$T_i := \{r : r \in S_i, i \in S_r\} \tag{4.2}$$

4. Define,

$$\alpha_i := \begin{cases} \text{The } \succ_i \text{ maximum element in } T_i & \text{if } T_i \neq \emptyset \\ \emptyset & \text{Otherwise} \end{cases} \tag{4.3}$$

   (S_i is the set of agents that i is looking at in a particular round. T_i is those agents that are also looking back at i, i.e. they are mutually interested in being matched. α_i is the best agent according to i who is also interested in being matched to i in this round.)

5. (a) Construct a directed graph $G = (N, E)$ with $(i, j) \in E$ iff $j = \alpha_i$. If $E = \emptyset$ move to STEP 6. Otherwise, proceed as below.

   (b) Identify all the strongly connected components of G. These are also all the cycles of G since every agent has an outdegree of at most 1. For any cycle, say, $m_1 \to w_3 \to m_3 \to w_1 \to m_2 \to w_4 \to m_1$, arbitrarily break it at the first edge or second to form a match between (m_1, w_3), (m_3, w_11) and (m_2, w_4), or between $(w_3, m_3), (w_11, m_2)$ and (w_4, m_1). Update the match $\hat{\mu}$ accordingly.

(c) Let $N := \{j : \hat{\mu}_j = 0\} \cap A$

(d) Go to Step 2.

6. Set

$$\hat{\beta}_i = \begin{cases} \beta_i & \text{if } \hat{\mu}_i > 0 \\ \min\{|A_i| + 1, \beta_i + 1\} & \text{if } \hat{\mu}_i = 0 \end{cases} \tag{4.4}$$

Definition 4.2. *The first instance when STEP 5 is run in $\mathbb{H}$, we draw a graph. This graph is called the Stage-1 graph of $\mathbb{H}$.*

4.5 Evaluation of $\hat{\mathbb{H}}$

$\hat{\mathbb{H}}$ is almost identical to $\mathbb{H}$ with a minor difference in STEP 6. Before running the $\hat{\mathbb{H}}$, one of the inactive agents is made active. Map $\hat{\mathbb{H}}$ does exactly the same operations as $\mathbb{H}$ only restricting attention to the active agents. The reason why this is specified as a different map is because $\mathbb{H}$ can potentially turn an active agent into inactive at the end of its execution. However, $\hat{\mathbb{H}}$ doesn't allow this. This is necessary in order to obtain a stable matching restricting attention to the active agents. Therefore, I will only state STEP 6, which is different from $\mathbb{H}$ below -

- Set

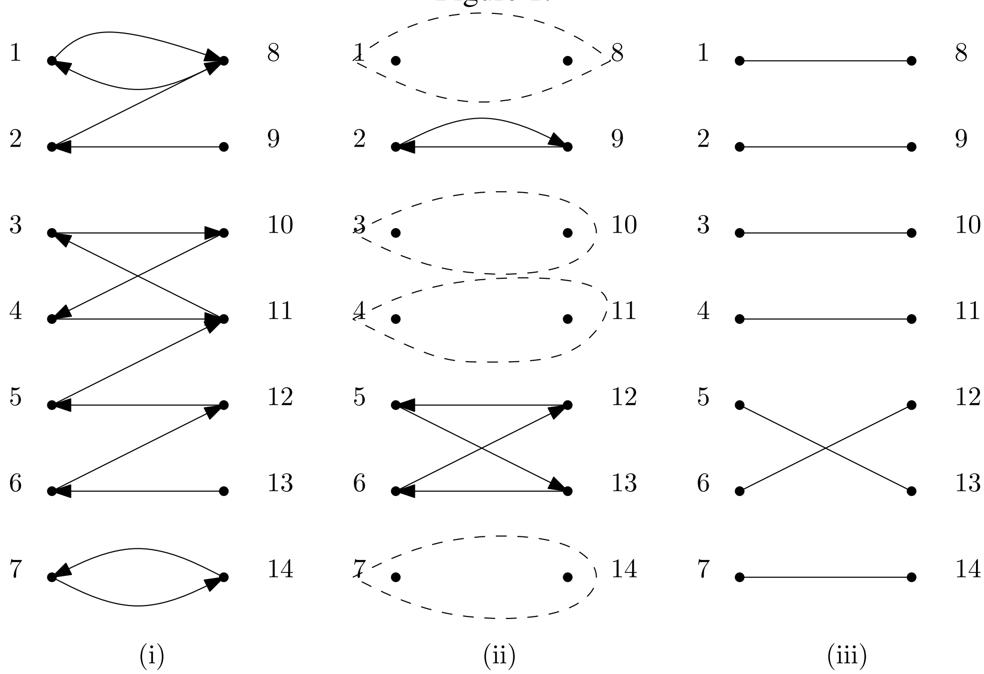
$$\hat{\beta}_i = \begin{cases} \beta_i & \text{If } \hat{\mu}_i > 0 \\ \min\{|A_i|, \beta_i + 1\} & \text{If } \hat{\mu}_i = 0 \text{ and } i \in N \\ \beta_i & \text{Otherwise} \end{cases} \tag{4.5}$$

4.5.1 Example of $\mathbb{H}$

It would be better to illustrate the map $\mathbb{H}$ with an example of a Graph. Suppose we have 7 agents on either side, say $(1, 2, .., 7)$ are men and $(8, 9, ..., 14)$ are women. And let us suppose that in some intermediate round the we evaluate the S_i, T_i and α_i from equations **??**, 4.2 and 4.3 to obtain the Stage-1 Graph. Suppose it looks like (*i*) in Figure 1.

Then, first we eliminate all the cycles. We break cycles of length 4 or more arbitrarily. Note that any cycle of length 4 or more can be broken to form a matching in only two ways. One way is to break it so that every man gets the woman he is pointing to or every woman gets a man she is pointing to. That is, we could think of this arbitrariness as just flipping a coin each time we have a cycle to break it one way or other. By doing that we obtain a Graph as in (*ii*). The dotted agents are absent in that graph but have been shown only to show what matches were formed in the first stage. 1 finds 8 the best and 8 finds 1 the best amongst the set of agents they can potentially be matched to in this round. (This set is given by T_i). $3 \to 10 \to 4 \to 11 \to 3$ is a cycle that is broken randomly to generate a matching $(3, 10)$ and $(4, 11)$. $(2, 14)$ is another match formed as they are the most preferred agents according to each other.

Figure 1:



Now, in the second stage, 2 can no longer point to 8 as 8 is matched in the first stage and is removed. So, in the graph in (*ii*), 2 points to, say, 9. Note that 2 might not point to any agents also. In this example, I have assumed that $9 \in S_2$ and $2 \in S_i$. That is, they find each other mutually acceptable *in this round*. Since this is a cycle of length 2, it is broken uniquely to get a match $(2, 9)$. Similarly 5 can no longer point to 11 and so points to 13 now, giving a 4 cycle- $5 \to 13 \to 6 \to 12 \to 5$. This is broken randomly to generate a match, say, $(5, 13), (6, 12)$.

The end result of this round $\mathbb{H}$ is final graph in (*iii*). Note that the operation in this round $\mathbb{H}$ could result in some instability. For example, look at (*i*). Here, if 11's preferences are $3 \succ_1 15 \succ_1 14$ then there will be an instability. This is because when we broke the cycle involving 4 agents. 11 was matched to 4. Moreover, 5 pointing to 11 despite finding 12 acceptable implies that $11 \succ_5 12$. Therefore, in $(4, 11)$ and $(5, 12)$ is an unstable match. However, there is no attention paid to stability in each round. In the next round, $5 \in T_{11}$ and $11 \in T_5$ and they have a potential to match. Various other things could happen in the next round but the idea is a blocking pair will be *discovered* eventually.

We allow each cycle to be broken arbitrarily *independently* of other cycles. Moreover, we also do not pay any attention towards stability.

5 Examples

We look at a simple example that we saw in 4.2 first. Then, we look at a slightly more involved example. Lastly, we look at an example where the algorithm fails to terminate. We will come back to that example after we present the full proofs of the procedure to be followed when the algorithm does not terminate.

5.1 A simple example

Here, we present the preferences of the entire market in one table. Agents m_1 to 4 can be thought of as men and a, b, c, d can be thought of as women.

m_1	m_2	m_3	m_4	w_1	w_2	w_3	w_4
w_1	w_2	w_3	w_4	m_2	m_3	m_4	m_1
w_2	w_3	w_4	w_1	m_3	m_4	m_1	m_2
w_3	w_4	w_1	w_2	m_4	m_1	m_2	m_3
w_4	w_1	w_2	w_3	m_1	m_2	m_3	m_4

The way to read this matrix is columns j denotes the preferences of agent j, ignoring the top row. That is, m_1s preferences are $w_1 \succ_{m_1} w_2 \succ_{m_1} w_3 \succ_{m_1} w_4$.

In the first two rounds nothing happens. There are no mutually proposing pairs and no matches are formed. That is,

$$T_i = \emptyset \ \forall i, \forall k \in M \cup W$$

In round 3,

$$T_{m_1} = \{w_2.w_3\}, T_{m_2} = \{w_3.w_4\}.T_{m_3} = \{w_4.w_1\}, T_{m_4} = \{w_1, w_2\} \tag{5.1}$$

$$T_{w_1} = \{m_3, m_4\}, T_{w_2} = \{m_4, m_1\}, T_{w_3} = \{m_1, m_2\}, T_{w_4} = \{m_2, m_3\} \tag{5.2}$$

Therefore,

$$\alpha_{m_1} = w_2, \alpha_{m_2} = w_3, \alpha_{m_3} = w_4, \alpha_{m_4} = w_1 \tag{5.3}$$

and

$$\alpha_{w_1} = m_3, \alpha_{w_2} = m_4, \alpha_{w_3} = m_1, \alpha_{w_4} = m_2 \tag{5.4}$$

And so, we have a cycle -

$$m_1 \to w_2 \to m_4 \to w_1 \to m_3 \to w_4 \to m_2 \to w_3 \to m_1$$

Cut the cycle arbitrarily at any point and we end up with

$\mu_1 = \{(m_1, w_2), (m_2, w_3), (m_3, w_4), (m_4, w_1)\}$ or $\mu_2 = \{(m_1, w_3), (m_2, w_4), (m_3, w_1), (m_4, w_2)\}$.

5.2 A Longer Example involving $\mathbb{H}$ and $\hat{\mathbb{H}}$

The preferences of men look as below -

m_1	m_2	m_3	m_4
w_2	w_3	w_4	w_1
w_3	w_4	w_3	w_4
w_1	w_1	w_1	w_2
w_4	w_2	w_2	w_3

And the preferences of women are -

w_1	w_2	w_3	w_4
m_4	m_3	m_1	m_2
m_1	m_4	m_4	m_4
m_3	m_1	m_2	m_3
m_2	m_2	m_3	m_1

Let us start with the algorithm. Observe that m_1 and w_2 appear on each other's top. Therefore, they will be matched in the first round and that match will never be broken.

Round 1 In the first round

$$T_i = \emptyset, \forall i \in \{m_1, m_2, m_3, w_2, w_3, w_4\}$$

$$T_{m_4} = \{w_1\}, T_{w_1} = \{m_4\}$$

Therefore,

$$\mu = \{(m_4, w_1)\}$$

Moreover,

$$\beta = (2, 2, 2, 1, 1, 2, 2, 2)$$

That is, $\beta_{m_1} = \beta_{m_2} = \beta_{m_3} = \beta_{w_2} = \beta_{w_3} = \beta_{w_4} = 2$. The reason is because these agents are unmatched in their first round and hence they increase their list by 1.

Round 2 Let us present a table for S_i, T_i and μ_i for each agent in this round. Recall that if an agent i is unmatched then $\mu(i) = 0$.

Table 1: Table of S_i, T_i, μ

Agent	S_i	T_i	μ
m_1	w_2,w_3	w_3	w_3
m_2	w_3,w_4	w_4	w_4
m_3	w_4,w_3	$\emptyset$	0
m_4	w_1	w_1	w_1
w_1	m_4	m_4	m_4
w_2	m_3,m_4	$\emptyset$	0
w_3	m_1,m_4	m_1	m_1
w_4	m_2,m_4	m_2	m_2

Therefore, at the end of this round, agents m_3 and w_2 are unmatched. So $\beta(m_3) = \beta(w_2) = 3$. β for all the other agents remains unchanged.

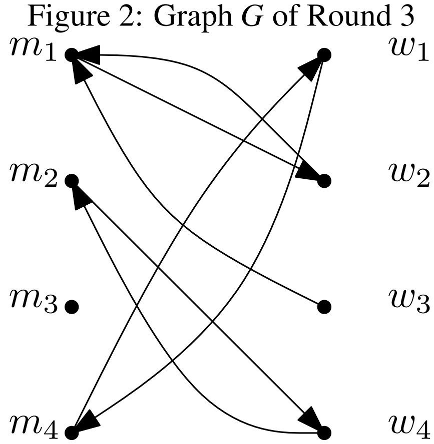
Round 3 Let us look at a similar table as above in this round. We also add β to this table instead of presenting it at the end of the round. It should be kept in mind that the β in the table corresponds to the updated β at the end of this round.

Table 2: Table of S_i, T_i, μ and β

Agent	S_i	T_i	μ	β
m_1	w_2,w_3	w_2,w_3	w_2	2
m_2	w_3,w_4	w_4	w_4	2
m_3	w_4,w_3,w_1	$\emptyset$	0	4
m_4	w_1	w_1	w_1	1
w_1	m_4	m_4	m_4	1
w_2	m_3,m_4,m_1	m_1	m_1	3
w_3	m_1,m_4	m_1	0	3
w_4	m_2	m_2	m_2	2

To see why the match in this round has changed, to be precise m_1 has left w_3, his match from Round 2 for w_2, look at the Stage-1 Graph of this round. Since m_1 is matched to w_3 in Round 2, m_1 and w_3 are in each other's T_i in this round. But in this round, w_2 proposes to m_1 and m_1 likes w_2 better than w_3. So in the graph m_1 points towards w_3.

Figure 2: Graph G of Round 3



At the end of this round, therefore, w_3 is unmatched and its β is updated to 3.

$$\mu^3 = \{(m_1, w_2), (m_2, w_4), (m_4, w_1)\}$$

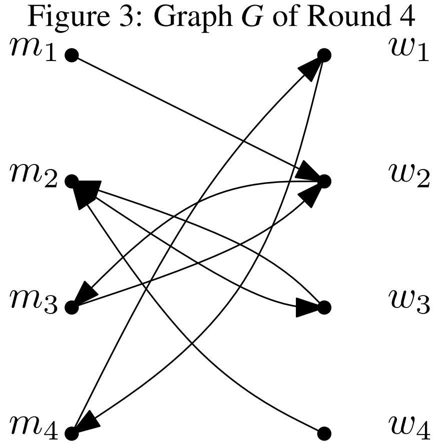
Round 4 Continuing as before, here is the table for Round 4.

Table 3: Table of S_i, T_i, μ and β

Agent	S_i	T_i	μ	β
m_1	w_2	w_2	0	3
m_2	w_3,w_4	w_3,w_4	w_3	2
m_3	w_4,w_3,w_1,w_2	w_2	w_2	4
m_4	w_1	w_1	w_1	1
w_1	m_4	m_4	m_4	1
w_2	m_3,m_4,m_1	m_1,m_3	m_3	3
w_3	m_1,m_4,m_2	m_2	m_2	3
w_4	m_2	m_2	0	3

Below is the graph for this round.

Figure 3: Graph G of Round 4



Since very soon we will start the cycling phase, it is important to note what μ^4 is.

$$\mu^4 = \{(m_2, w_3), (m_3, w_2), (m_4, w_1)\}$$

Round 5 And we continue the process.

Table 4: Table of S_i, T_i, μ and β

Agent	S_i	T_i	μ	β
m_1	w_2,w_3,w_1	w_3	w_3	3
m_2	w_3	w_3	0	3
m_3	w_4,w_3,w_1,w_2	w_2,w_4	w_4	4
m_4	w_1	w_1	w_1	1
w_1	m_4	m_4	m_4	1
w_2	m_3	m_3	0	4
w_3	m_1,m_4,m_2	m_1,m_2	m_1	3
w_4	m_2,m_4,m_3	m_3	m_3	3

Therefore,

$$\mu^5 = \{(m_1, w_3), (m_3, w_4), (m_4, w_1)\}$$

Round 6 Continuing this way, in this round, we have -

Table 5: Table of S_i, T_i, μ and β

Agent	S_i	T_i	μ	β
m_1	w_2,w_3	$w_3.w_2$	w_2	3
m_2	w_3, w_4, w_1, w_2	w_4	w_4	3
m_3	w_4	w_4	0	5
m_4	w_1	w_1	w_1	1
w_1	m_4	m_4	m_4	1
w_2	m_3, m_4, m_1, m_2	m_1	m_1	4
w_3	m_1	m_1	0	4
w_4	m_2,m_4,m_3	$m_2 m_3$	m_2	3

Thus, in this round, agent m_3 has become inactive.

$$\mu = \{(m_1, w_2), (m_2, w_4), (m_4, w_1)\}$$

Round 7 Similarly, in this round, w_3 fails to break the previous match and becomes inactive. Therefore, we now have a match μ obtained as before with m_3, w_3 being the two inactive agents. Now, we run the map $\hat{\mathbb{H}}$ by making agents sequentially active. If we make w_3 active first, then it breaks the existing matching (since m_2 prefers w_3 to w_4 to yield,

$$\mu = \{(m_1, w_2), (m_2, w_3), (m_4, w_1)\} \tag{5.5}$$

It can be seen that μ is stable with respect to the active agents. Hence, we make m_3 active now to obtain,

$$\mu_1 = \{(m_1, w_2), (m_2, w_4), (m_3, w_3), (m_4, w_1)\} \tag{5.6}$$

Alternatively, if we make m_3 active first then, it breaks the matching μ obtained before to yield,

$$\mu = \{(m_2, w_4), (m_3, w_2), (m_4, w_1)\} \tag{5.7}$$

Following this, we make agent w_3 active to obtain,

$$\mu_2 = \{(m_1, w_3), (m_2, w_4), (m_3, w_2), (m_4, w_1)\} \tag{5.8}$$

It can be seen that μ_1, μ_2 are correspondingly the Men-Optimal and Women-Optimal stable matchings and those are the only two stable matchings in this market. Interestingly, the procedure of Equitable Stable Matching due to Romero-Medina [11] obtains only μ_1.

5.3 An example highlighting the difference with Random-Priority Mechanism

The following example is from Roth-Sotomayor [12] The preferences of men look as below -

m_1	m_2	m_3	m_4
w_1	w_2	w_3	w_4
w_2	w_1	w_4	w_3
w_3	w_4	w_1	w_2
w_4	w_3	w_2	w_1

And the preferences of women are -

w_1	w_2	w_3	w_4
m_4	m_3	m_2	m_1
m_3	m_4	m_1	m_2
m_2	m_1	m_4	m_3
m_1	m_2	m_3	m_4

There are ten stable matchings in this market.

	w_1	w_2	w_3	w_4
1	m_1	m_2	m_3	m_4
2	m_2	m_1	m_3	m_4
3	m_1	m_2	m_4	m_3
4	m_2	m_1	m_4	m_3
5	m_3	m_1	m_4	m_2
6	m_2	m_4	m_1	m_3
7	m_3	m_4	m_1	m_2
8	m_4	m_3	m_1	m_2
9	m_3	m_4	m_2	m_1
10	m_4	m_3	m_2	m_1

The median stable matching of Sethuraman and Teo [15] is either matching 4 or 7. The Random-Priority Mechanism due to Ma [8] generates any matching *except* $4, 5, 6$ and 7. The algorithm I propose can produce any matching out of $4, 5, 6, 7$.

5.4 Difference between Center Stable Matching and The Proposed Algorithm

m_1	m_2	w_1	w_2
w_1	w_2	m_2	m_1
w_2	w_1	m_1	m_2

Consider the market above. The two stable matchings are, $(m_1, w_1), (m_2, w_2)$, and $(m_1, w_2), (m_2, w_1)$. The proposed algorithm, the *Center Stable Matching* and the *Median Stable Matching* all yield the same answer here. Now consider three identical copies of the same market packed into one market.

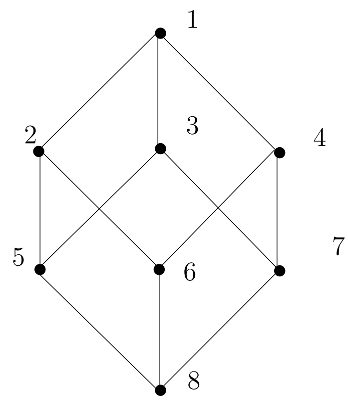
W_1	W_2	W_3	W_4	W_5	W_6
M_2	M_1	M_4	M_3	M_6	M_5
M_1	M_2	M_3	M_4	M_5	M_6

The set of stable matchings is -

Matchings	M_1	M_2	M_3	M_4	M_5	M_6
μ_1	1	2	3	4	5	6
μ_2	2	1	3	4	5	6
μ_3	1	2	4	3	5	6
μ_4	1	2	3	4	6	5
μ_5	2	1	4	3	5	6
μ_6	2	1	3	4	6	5
μ_7	1	2	4	3	6	5
μ_8	2	1	4	3	6	5

The Hasse diagram looks as follows -

The median matching is only μ_1 and μ_8. The *Center Stable Matching* are μ_2 to μ_7. The proposed algorithm can yield any matching between μ_1 to μ_8 as a solution.



6 Results

Remark 1. *For any graph obtained in the execution of $\mathbb{H}$, if a node has an incoming edge then it must have an outgoing edge.*

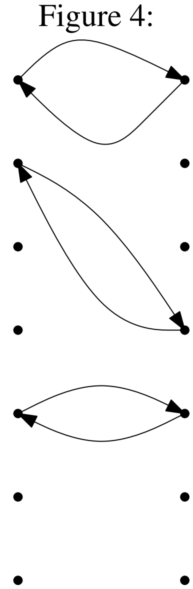
Remark 2. *If, for any node i, $deg^{-}(i) > 0$ then $deg^{+}(i) = 1$.*

Remark 3. *For any i, $deg^{+}(i) \leq 1$.*

Definition 6.1. *For any Stage-1 graph (Defn. 4.2), call it of Type-1 if it consists of only cycles of length 2 and singletons.*

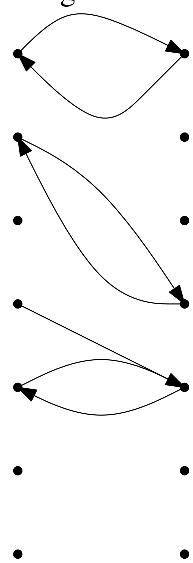
For example, Figure 4 is of Type-1.

Figure 4:



While Figure 5 is NOT of Type-1

Figure 5:



The proof of how the algorithm terminates and outputs a stable match on all preference profile is carried out this way -

1. If the algorithm terminates in STEP 3 then it produces a stable match.
2. Step 3 is always reached.
3. If the algorithm goes to STEP 4 and 5, then the critical step is 5(c). We then show that 5(c) always terminates for every j. And therefore the algorithm always terminates.
4. The output from STEP 5 is a stable matching.

Remark 4. *It is important to note that since $\mathbb{H}$ breaks cycles arbitrarily, it is not immediately clear that if $(\mu, \beta) = \mathbb{H}(\mu, \beta) \Rightarrow (\mu, \beta) = \mathbb{H}(\mathbb{H}(\mu, \beta))$.*

Lemma 6.1. *If round k is a terminal round, then the Stage-1 Graph of $\mathbb{H}(\mu, \beta)$ must be of Type-1.*

Proof. In other words, what we need to prove here is that if the Graph is not of Type-1 then for some agent i, either S_i is updated or β_i is updated. We will prove that S_i is updated. Since μ_i and β_i jointly determine S_i, if β_i is not updated then S_i being updated implies that μ_i must have changed.

Define,

$$V := \{j : \mu(j) > 0\}$$

Then, $\beta_i = |A_i| + 1 \ \forall i \notin V$. This is because, since μ is a terminal matching, in the previous round, when $\mathbb{H}$ was applied on μ, β we obtained μ, β again. So, for the unmatched agents, β can remain the same only if they are inactive.

Suppose $G = (N, E)$ is not of Type-1. $\Rightarrow E$ (the set of edges) is non-empty. Since our bipartite graph is such that if $deg^{-}(i) > 0 \Rightarrow deg^{+}(i) = 1$, set of edges being non-empty means the only possible configurations are cycles of even length with or without incoming edges to them. Since our assumption is that G is not of Type-1, $\exists$ a cycle $C = \{a_1, ., a_n\}$ with or without some incoming edges to some nodes.

Case 1: Length of $C > 2$.

Our algorithm entails breaking this cycle arbitrarily. Let's assume we break it at a_1, a_2. That is, we match (a_1, a_n) and (a_2, a_3). Since $a_1 \to a_2 \to a_3 \to ...a_n \to a_1$, we have that $a_3 \succ_{a_2} a_1$.

Since this is a terminal round, it must be the case that $\mu^k = \mu^{k+1}$. Therefore, we must have $\mu^k(a_1) = a_n$ and $\mu^k(a_2) = a_3$. Since the cycle is $a_1 \to a_2 \to a_3 \ldots \to a_n$, we have that $a_3 \succ_{a_2} a_1$. But then, since we had $\mu^k(a_2) = a_3$ to begin with, $a_1 \notin S^k_{a_2}$. Therefore, edge (a_1, a_2) is not possible.

Therefore, we cannot have a cycle of length > 2.

Case 2: Length of $C = 2, \forall C$.

Given that our assumption is that G is not of Type-1, this means that $\exists C$ of length 2, say (a_1, a_2) with an edge, say $a_3 \to a_1$.

If we have $\mu^k(a_1) = a_3$, then at the end of this round, (since a cycle of length 2 will be broken in a unique way) we will have $\mu^{k+1}(a_1) = a_2$. Therefore, this cannot be a terminal round.

If $\mu^k(a_1) = a_2$ then, given that $a_2 \succ_{a_1} a_3$ edge (a_3, a_1) is not possible in this stage.

That is, if there is a cycle of length 2 then it cannot have any incoming edge from outside.

□

Lemma 6.1 can be restated as, *If a round k is a terminal round, then the Stage-1 Graph of it must have only two-cycles and isolated points.*

Lemma 6.2. *At the end of STEP 2, every agent's T_i contains at most one agent.*

Proof. Suppose $\exists\ i \in M$ such that $j, k \in T_i$. Let, $j = \alpha_i$. By Type-1 nature, $i = \alpha_j \Rightarrow \mu(i) = j, \mu(j) = i$. Moreover, since the matching remains unchanged in the terminal round, this must have been the matching to begin with. But then, $j \succ_i k, \mu(i) = j \Rightarrow k \notin S_i$. Hence proved. □

Proposition 6.1. *If the algorithm terminates with $\Gamma = M \cup W$ then the matching μ obtained in the last round is stable.*

Proof. Suppose not.

Say, round k is the terminal round. Let the round input be (μ^k, β^k).

Therefore, if

$$(\mu^{k+1}, \beta^{k+1}) = \mathbb{H}(\mu^k, \beta^k)$$

then, $\mu^{k+1} = \mu^k$.

Suppose, $\exists$ a blocking pair $(i.j)$. So, $i \succ_j \mu^k(j)$ and $j \succ_i \mu^k(i)$. Since $j \in T^k_i$ and $i \in T^k_j$, $deg^{+}(i) = deg^{+}(j) = 1$.

By lemma 6.1, we know that the stage graph of $\mathbb{H}$ applied on the terminal round's μ^k, β^k would be of Type-1.

That is, $\exists$ an agent m such that (i, m) form a two-cycle in the Stage-1 Graph. Moreover, since Stage-1 Graph is drawn by each agent pointing to its top agent in T_i^k, i cannot be pointing to $\mu^k(i)$ as $j \in T_i^k$ and $j \succ_i \mu^k(i)$. By Type-1 nature of the graph, i will be matched to m or someone that i prefers to m. Since $m \succ_i \mu^(i)$.

$\Rightarrow \mu^{k+1}(i) \neq \mu^k(i)$.

A contradiction.

$\square$

For the sake of Lemma to appear next, consider any stable matching μ. Define β as follows -

$$\beta_i = \begin{cases} \text{Rank of agent } \mu(i) \text{ according to } \succ_i & \text{if } \mu(i) > 0 \\ |A_i| & \text{if } \mu(i) = 0 \end{cases} \tag{6.1}$$

Recall that A_i was the set of all the acceptable agents for i.

Lemma 6.3. $\mathbb{H}(\mu,\beta) = (\mu,\beta)$*, where* μ *is some stable matching and* β *is defined accordingly as per the equation above.*

Remark 5. *That is, Lemma 6.3 says that if in a particular round, the algorithm encounters a stable matching such that all the unmatched agents are at the end of their proposal sets, then the algorithm terminates at that matching.*

Proof. Since μ is stable, $\nexists$ a pair (i, j) such that $j \succ_i \mu(i)$ and $i \succ_j \mu(j)$.

Therefore, for any $j \in \{k : k \succ_i \mu(i)\}$, $\mu(j) \succ_j i$.

Therefore, $j \notin T_i$.

So,

$$T_i = \begin{cases} \mu(i) & \text{if } \mu(i) > 0 \\ \emptyset & \text{if } \mu(i) = 0 \end{cases}$$

So, the stage-1 graph will be of Type-1. We will break the cycles uniquely with $\hat{\mu(i)} = \mu(i)$ $\forall i$ such that $\mu(i) > 0$.

Moreover, since for the unmatched agents, $\beta(i) = |A_i|$, the β_i is not updated for any agent.

Therefore, $\hat{\mu} = \mu$.

So, $\mathbb{H}(\mu,\beta) = (\mu,\beta)$. $\square$

Proposition 6.2. *STEP* 3 *of the algorithm is always reached.*

Proof. In every round, the map $\mathbb{H}$ either updates a matching or changes the β. When a matching is updated, one of the following things happen -

- Unmatched agents remain unmatched and their β increases if they are active. And they may potentially become inactive.
- Matched agents get a better match.
- Matched agents get unmatched and their β increases.

Note that, any agent i cannot move from his current match to someone worse than it without being unmatched and therefore increasing his β. There are at most n different agents any agent can be matched to. As proved in Lemma 6.1, if the (μ,β) is a terminal input then the graph is Stage-1. Therefore, every agent's match can change at most $n^2 + 1$ times. This is because, an agent can go from his worst to best match, which takes n steps) and then must come back to unmatched state before starting this process again. Every time the agent becomes unmatched his β increases. An agent becomes inactive once his(her) β equal to $|A_i| + 1 \leq n + 1$. In particular, any agent can obtain his least preferred agent at most once. Since, after this the agent can only improve and if he becomes unmatched, his β becomes $|A_i| + 1$ and the agent becomes inactive. Therefore, an agent can be unmatched at most $n + 1$ times. So, every agent can get his rank n agent at most once, rank $n - 1$ agent at most twice and so on. Therefore, an agent's position may change at most $\frac{n(n+1)}{2}$ times. Moreover, there are $n + 1$ rounds of increasing his β to be able to complete this process. After changing his match $\frac{n(n+1)}{2}$ times, the agent becomes inactive. Therefore, after $\frac{n(n+1)}{2} + n$ changes, an agent cannot remain active. Since there are $2n$ agents in all, after $2n[\frac{n(n+1)}{2} + n] = n^3 + 2n^2$ rounds, μ,β pair cannot change. Therefore, after $n^3 + 2n^2$ rounds, STEP 2 must end. □

We have therefore obtained two things -

- The algorithm always goes to STEP 3. That is, STEP 2 cannot last forever.
- If the set of matched agents at the end of STEP 2 is all the men and women then the matching obtained is stable.

Therefore, all that remains to show now is STEP 5 terminates and produces a stable matching.

Definition 6.2. *A matching μ is stable restricting attention to set S if $\nexists$ a blocking pair (i, j) such that $i, j \in S$.*

Lemma 6.4. *If A is the set of active agents at the end of STEP 3, then the matching μ obtained from STEP 2 is stable restricting attention to A.*

Proof. Suppose *exists* a blocking pair $(i, j) \in A$.

$\Rightarrow i \in T_j, j \in T_i$. Therefore, $\alpha_i = j$, since $|T_i| \leq 1$ according to lemma 6.2. Therefore, $j = \mu(i), i = \mu(j)$. So, (i, j) cannot be a blocking pair as they are already matched to each other. □

Once we enter STEP 5, the algorithm first looks at all the inactive agents. Picks one agent randomly and makes him (her) active. Thereafter, the algorithm runs step $5(c)$ until there is convergence. Once we have convergence, the procedure is repeated by making a new agent active. Therefore, our task is to first prove that we do in fact get convergence. Secondly, what we eventually obtain is a stable matching. The first lemma below proves the first part.

For the sake of the lemma, consider a set $S \subseteq M \cup W$ and a matching μ such that $\mu(i) > 0 \Rightarrow i \in S$ (all the matched agents are in S). Moreover, suppose that μ is stable restricting attention to S. Consider any β such that $\beta_i \leq |A_i| \Leftrightarrow i \in S$. That is, an agent is active iff (s)he is in S.

Lemma 6.5. *Consider an agent $i \notin S$ (wlog). Let $\beta_i = |A_i|$. That is, we make the agent i active. Then, repeated application of $\hat{\mathbb{H}}(\mu, \beta)$ terminates in at most n^2 iterations.*

Proof. Suppose $i = w$. Only two things can happen upon adding w to the set of active agents S. Either there is a blocking pair or not. Since μ had no blocking pair restricting attention to S, any blocking pair must involve w. That is, the Stage-1 graph will either have some men pointing to w as they like w better than their current match or no agent finds w attractive. In the first case the graph will not be of Type-1, while in the second case the graph will be of Type-1. So, Stage-1 graph of $\hat{\mathbb{H}}$, if there is a blocking pair would as one of the two pictures below.

In either case, the graph will only have two cycles and upon breaking them, every man is weakly better off. At least one man is strictly better off due to getting matched to w. And since w gets matched to some man m, $m's$ earlier match, if any, is now unmatched. In every iteration of $\hat{\mathbb{H}}$, at most one woman goes from matched to unmatched. In this process a man she was matched to initially strictly improves. If no woman goes from matched to unmatched then either an unmatched woman gets matched to a single man or a woman remains unmatched. In either case, we have convergence. If one man strictly improves in every round, this can happen only finite number of times and hence will stop in at most n^2 rounds. □

It is important to note that for this process to work it was necessary to define $\hat{\mathbb{H}}$ which differed from $\mathbb{H}$ in one way that no active agent could become inactive in $\hat{\mathbb{H}}$. Therefore, repeated application of $\hat{\mathbb{H}}$ terminates keeping all the agents in S and a newly added agent active.

Lemma 6.6. *Let $\hat{\mu}$ be the matching obtained by repeated application of $\hat{\mathbb{H}}$ on (μ, β) with S as the initial active set and i as the newly added agent. Then, $\hat{\mu}$ is stable restricting attention to $S \cup \{i\}$.*

Proof. Firstly, an almost identical proof of Lemma 6.1 applies for termination of $\hat{\mathbb{H}}$ to obtain that whenever $\hat{\mathbb{H}}$ terminates, the Stage-1 graph is of Type-1. Type-1 nature gives us Lemma 6.2 to establish that $|T_i| \leq 1 \; \forall i$. Therefore, we cannot have a blocking pair in $S \cup \{i\}$. □

We can combine Lemma 6.1, Proposition 6.1, Lemma 6.5 and Lemma 6.6 to present the main theorem.

Theorem 6.3. *Algorithm 1 outputs a stable match for all preferences profiles.*

Proof. Proof is simply putting together all the results obtained. Lemma 6.2 proves that STEP 2 always ends. If STEP 2 ends with all agents being active then Proposition 6.1 proves that the output is a stable matching. If some agents are inactive, then we enter STEP 4 and 5. Lemma 6.4 establishes that the input to STEP 4 is a stable matching restricting attention to the set of active agents. STEP 5 has a repeated application of $\hat{\mathbb{H}}$ by changing one inactive agent to active sequentially. Lemma 6.5 and 6.6 tell us that every time an agent is made active, repeated application of $\hat{\mathbb{H}}$ terminates to give a stable matching restricting attention to the set of new active agents. Moreover, $\hat{\mathbb{H}}$ never turns an active agent into inactive. Therefore, this procedure of adding inactive agents one by one terminates to output a stable matching. □

7 Computational Compleixity

The *Median Stable Matching* by Sehutraman and Teo [14] is a very appealing procedure when fairness concerns matter. However, a practical problem is to compute the median matching one has to compute the entire set of stable matchings which is a hard problem, computationally. The proposed algorithm has a complexity of $O(n^5)$.

Theorem 7.1. *Algorithm 1 can be executed in $O(n^7)$ time.*

Proof. As argued in Lemma 6.2, we need at most kn^3 rounds to reach STEP 3. Each round has an execution of map $\mathbb{H}$. $\mathbb{H}$ involves computing T_i to construct the graph G. T_i can be computed in $O(n^3)$. Following the construction of the graph, we need to enumerate the cycles which is the same as enumerating all the strongly connected components. This can be done using Tarjan's algorithm in $O(|V| + |E|) = O(n)$. The process of eliminating cycles can be repeated within a particular execution of $\mathbb{H}$. Every iteration removes at least 2 agents by forming a match between them. Therefore, we will need to repeat this at most n times. Each iteration involves reconstructing the graph that can be done in $O(n^3)$ time. Therefore, a particular execution of $\mathbb{H}$ can be done in $O(n^4)$ time. Therefore, we can reach STEP 5 in $O(n^7)$ time. STEP 5 involves turning each inactive agent into active. Therefore, we will need to repeat this at most $2n$ times. Each execution runs $\hat{\mathbb{H}}$ until we get convergence. As proved in Lemma 6.5, operation takes at most n^2 rounds. Each execution of $\hat{\mathbb{H}}$ can be done in $O(n^4)$ time. Therefore, an operation following making one agent active from inactive can be done in $O(n^6)$ time. Since there are at most $2n$ agents to be made active, STEP 5 can be finished in $O(n^7)$ time. Hence proved. □

8 Fairness Properties

Part of the motivation for the algorithm in the introduction was to design a *fair procedure*. As mentioned before, various randomized procedures such as those in Klaus and Klijn (2006) [5] (Employment by Lotto proposed by Aldershof et al [1] or Random Order Mechanism by Ma [8] that is based on Roth and Vande-Vate (1990)[13] exist to achieve ex-ante fairness. However, a key distinction between these procedures and the one I propose is the nature of the procedure itself where, due to both sides proposing in every round, the outcomes tend to be at the centre than at the extremes. Through the various examples presented, we can see this distinction more clearly.

One possible and perhaps necessary approach towards studying fairness in the context of stable matchings is axiomatic. The two most natural axioms that can be motivated purely from fairness standpoint would be *Gender Neutrality* and *Permutation Invariance.* That is, any *fair* procedure should not distinguish between Men and Women and should also be immune to relabeling men or women. As Gokturk and Masarani (1989)[10] showed that there does not exist a deterministic stable matching mechanism that satisfies these two properties. Therefore, even restricting ourselves to these two seemingly necessary requirements for fairness we ought to have randomized mechanisms. One possible randomized mechanism could be to have a uniform lottery on the entire set of stable matchings. Another mechanism could be to select one of the two Gale-Shapley outcomes with equal probability. Both these mechanisms would satisfy *Gender Neutrality* and *Permutation Invariance.*

To present the axioms, let's first denote the support of the algorithm as Ψ. Since the algorithm involves breaking cycles in each round, depending on how the cycles are broken or the order we may obtain different matchings. Let the entire set of such matchings that can be obtained by some sequence of breaking the cycles be Ψ.

$$\Psi(P_M, P_W) := \{\mu : \mu \textit{ can be obtained through the algorithm.}\} \tag{8.1}$$

Let Γ denote any stable matching correspondence.

Axiom 8.1. ***Gender Neutrality:*** $\Gamma(P_M, P_W) = \Gamma(P_W, P_M)$.

Axiom 8.2. ***Permutation Invariance:*** $\Gamma(\pi(P_M), \pi(P_W)) = \pi(\Gamma(P_M, P_W))$*, where π denotes any permutation of men and women.*

Axiom 8.1 says that the the rule should not change the outcome by rebelling men as women or vice-versa. Axiom 8.2 says that for any permutation of men and women, the outcome of the rule should not change. Both these axioms are very natural from the purview of fairness and it can be easily seen that Algorithm 1 satisfies both these axioms.

9 Conclusion

In this article I focus on developing a new procedure that does not distinguish between men and women. I propose a new procedure that has an appealing feature of both sides actively proposing in each round. The most commonly used procedure in two-sided matching markets is Gale and Shapley's (1962)[4] Deferred Acceptance Algorithm. The DAA has only one side proposing in each round and the other side accepting or rejecting proposals. In contrast, here in each round, both sides of the market make proposals and only mutually acceptable agents are matched in each round.

The algorithm, motivated from fairness concerns, can be run in polynomial time. This procedure should be viewed as an addition to the existing procedures that are gender neutral. In the quest to obtain stable matchings that treat both the sexes equally, Sethuraman and Teo (2006)[15] proposed the *Median Stable Matching.* A practical problem with the *Median Stable Matching* is we need to compute the entire set of stable matchings in order to obtain the *Median Stable Matching.* In contrast with the other proposed rules satisfying Gender Neturality and Permutation invariance (Ma (1996) [8], Romero-Medina(2005)[11] this algorithm does not require an ordering over agents to start the algorithm. Both the procedures need an explicit priority order over agents in order to begin the algorithm. Moreover, as seen in examples in section 5.3, there is a ground to argue that the proposed algorithm produces fairer outcomes.

A natural addition to the procedure is to characterize the matching it obtains. Through various examples, it is seen that the algorithm often selects outcomes that are different from the DAA outcomes. Despite some randomness involved in breaking the cycles, this procedure is different from selecting a random stable matching or a random matching and eliminating blocking pairs proposed by Roth and Vande (1990). My conjecture is that the support of the matchings that can be achieved through the algorithm proposed is a sublattice that always contains the Rawlsian stable matching.

Thus, this algorithm, should primarily be viewed as an attempt to address a remark by Knuth mentioned in the introduction. In contrast with the other procedures that address this concern, by imposing an ordering over agents, this is the first procedure wherein both sides participate in each round, beyond just accepting or rejecting offers. In practice, due to its ex-ante fairness and efficiency in computing, this algorithm may be a good alternative to the currently existing procedures.

References

[1] Aldershof B., Carducci O.M. and Lorence D.C., *Refined inequalities for stable marriage*, Constraints, 4:281-292, 1999.

[2] Cheng C., *Understanding the Generalized Median Stable Matchings*, Algorithmica, 58:1, 34-51, 2010

[3] Cheng C., McDermid E. and Suzuki I., *The center stable matchings and the centers of cover graphs of distributive lattices,* , ICALP, 2011

[4] Gale D. and Shapley L.S., *College admissions and the stability of marriage*. American Math. Monthly, 69: 9-15, 1962.

[5] Klaus B. and Klijn F., *Procedurally fair and stable matching*, Economic Theory, 27:431-447, 2006

[6] Klaus B. and Klijn F., *Median stable matching for college admissions*, International Journal of Game Theory, 34:1-11, 2006

[7] Knuth D., *Marriages Stables*, Montreal: Les Presses de l Universite de MOntreal, 2,3, 1976.

[8] Ma J. *On randomized matching mechanisms*, Economic Theory, 8:377-381, 1996.

[9] Manjunath V. and Turhan B., *Two school systems, one district: What to do when a unified admissions process is impossible* , Games and Economics Behaviour, 2015

[10] Masarani F. and Gokturk S.S, *On the existence of fair matching algorithms*. Theory and Decisions, 26: 305-322, 1989

[11] Romero-Medina, A., *Equitable selection in bilateral matching markets*, Theory and Decisions, 58:305-324, 2005

[12] Roth A.E. and Sotomayor M.A.O., *Two-sided matching: A study in game-theoretic modeling and analysis*, Econometric Society Monograph Series, New York: Cambridge University Press, 1990.

[13] Roth A.E. and Vande Vate J.H. *Random paths to stability in Two-Sided matching*, Econometrica, 58:1475-1480, 1990.

[14] Sethuraman J., Teo C.P. and Qian L., *Many-to-one stable matching; Geometry and Fairness*, Mathematics of Operations Research, 31(3):581-596,2006.

[15] Teo C.P. and Sethuraman J., *The geometry of fractional stable matchings and its applications*, Mathematics of Operations Research 23:874-891,1998.